

How to / Trackmania / Trackmaking / Blockmix / Hex-Edit Author : NL_pwf , TStarGermany (Translations: Natzor , -DexM-)

Introduction

What is blockmixing or hex editing ? Well...it's combining 2 or multiple parts that normally can't be combined and placing blocks on places where they normally can't be placed.

The results are trackworks and scenery, which feature outstanding looks or functionality, not provided by the internal track editor.

Basic Blockmixing techniques are to be learned easily, but then again, more sophisticated Blockmixing requires lots of experience and abstraction... We'll see how far you can get :-)

Part 1 : Structure of a Trackfile (GBX)

Trackmania driving tracks are stored in the .GBX file format, Nadeo's own file format ... A .GBX file contains lots of different data, written in "computer code", lots of them encrypted and/or compressed, so we need a "[Hex-Editor](#)" to look at the file's content (or change it) properly.



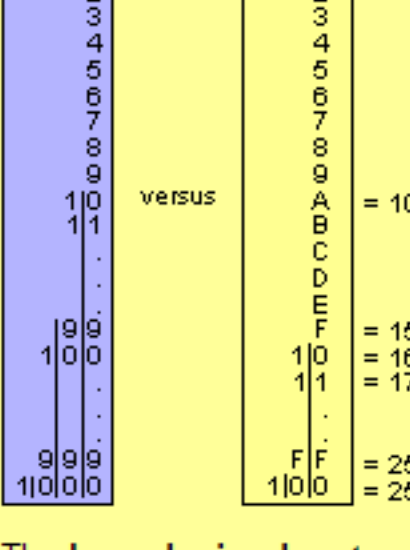
In the case of a track file, the GBX file roughly contains this information:

- * general track information (like track name, author name , validation time, password etc.pp) * tile placement (all your roadworks)
- * mediatracker content
- * file dependencies (locations/URLs of signs, mods, music etc...)
- * a thumbnail screenshot in JPG format and others...

different parts are highlighted with different colours in the picture on the left...

As you can see, every piece of information is displayed in hexadecimal code on the left...

Decimal and Hexa-decimal



Usually we don't think much about it, but the way the "write down" our numbers every day is pretty complicated!

In our "normal" **decimal system**, we count numbers from 0-9 ... if we want to continue counting, we add 1 "register" to the left, raise its value by "1" and continue counting on the right side again...

For example:
576 is actually 5/7/6 (100*5 + 10*7 + 1*6)
To be more precise:
10²*5 + 10¹*7 + 10⁰*6

The **hexadecimal system** does the same, except it does not only include 0-9 (like decimal), but 0-9 and A-F ... that means, 16 counting ciphers until it has to add 1 "register" on the left and start counting again...

What for ? In short terms, machine code basically consists of "0" and "1", two different electric signals, being on/off, a computer doesn't understand anything but this.

While "0" and "1" are called "bits", a hexadecimal cipher represents 2⁴ bits ... a full "byte" is represented by 2⁸ bits = 2 hexadecimal ciphers (=the possibility to set 256 values).

To edit the bytes in our track file, we must be able to read & write these bytes.

What if ?

If a value we want to set exceeds "256" ? Then you need an additional byte attached to the one byte you already have ... multi-byte values ... but don't worry, TM does not use multi-byte values :-)

What heximal value represents what "real" character ?

The bytes / hex code on the left side is mostly represented by ASCII- characters on the right side. Not all bytes are displayable with [ASCII](#), so sometimes you will find empty gaps on the right.

This was a pretty short introduction, but you should now have an idea what the hexadecimal values in the hex editor are expressing. Of course GBX files can also contain driving replays or 3D car models etc..., but we will only talk about the track files here to teach you some mixing.

How tiles are placed and stored



In the case of the stadium, you have 32*32*32 cells to place your tiles. Each tile you place has a name, a coordinate (length, height and depth, going from value "0-31").

In case of the height (Y in the picture), the grass ground itself has value "0", everything above the ground starts with "1"

In addition to the tile's name and XYZ **position**, also its rotation and variation are stored in the track file (**variations** occur, wenn the same part has different shapes, e.g. when being connected to the floor or another tile next to it).

Part 2 : Manual Blockmixing

Not so long ago, people had to do blockmixing with the Hex-Editor. If you don't want to know about it (although you should), you can directly jump to [Part 3](#).

To keep a track's filesize small, Nadeo's GBX file format compresses all information about what tile is placed where on the track, making it virtually impossible to find all necessary information for 1 single selected piece you wanted to change.

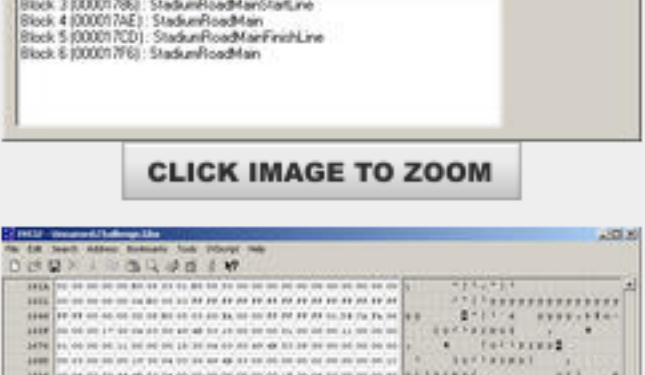


You place a grass checkpoint at **X:17/ Y:1 / Z:17** ... right next to the turbo, on which you want to mix the checkpoint on...

So the task is to move the checkpoint from **X:17 to X:16**. Everything else can remain unchanged.



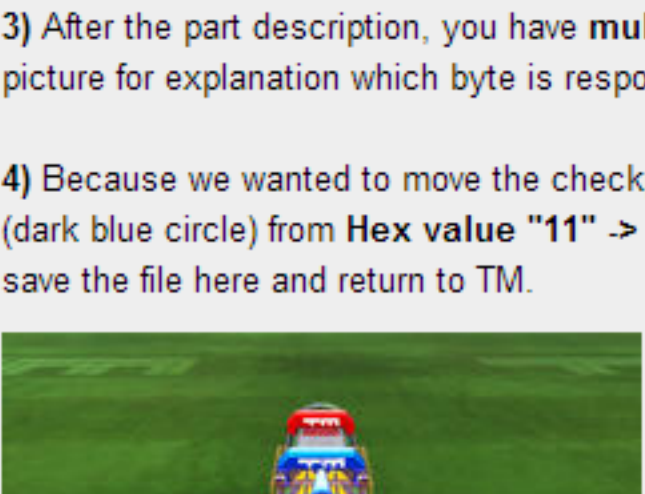
A tool called "**ReCompressor**" is able to de-compress this information, save the file again in a deflated version and **list all available blocks with the Hex adress**ing, where they can be found.



Now you could open the decompressed track with your Hex Editor:

- 1) You went to the **Hex adress 00001760** (which you know from the ReCompressor listing), symbolised by a red square.
- 2) To the right side, you can see where the "**StadiumGrassCheckPoint**" (yellow) actually begins.

- 3) After the part description, you have **multiple bytes**, responsible for the part's **rotation, location and variation**. See picture for explanation which byte is responsible for what.
- 4) Because we wanted to move the checkpoint from **X:17 -> X:16**, we will change exactly this byte in the Hex Editor (dark blue circle) from **Hex value "11" -> "10"** (representing the "real" values "17" -> "16", our coordinates). Now we can save the file here and return to TM.



When opening the track again in TM, we will see, that the checkpoint has moved 1 cell, directly onto the turbo part :-)

Now we can continue building.

Luckily, TM will use its space saving compression method, as soon as we save the track file again.

This is roughly, how manual block mixing works. Before the ReCompressor tool was released, things would become much more complicated as soon as the GBX compression algorithm began to group, hide and just leave out bytes for certain tiles.

Even today, you shouldn't use ReCompressor anymore, if you wanted to do manual Hex-Editing, because it had some errors while processing the files.

Today, you should use the ChallengeEdit tool for decompressing, described in the next chapter.

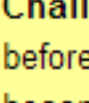
Part 3 : Modern Blockmixing

Luckily, Blockmixing has become very fashionable, so today we have some easier methods to mix our tiles.

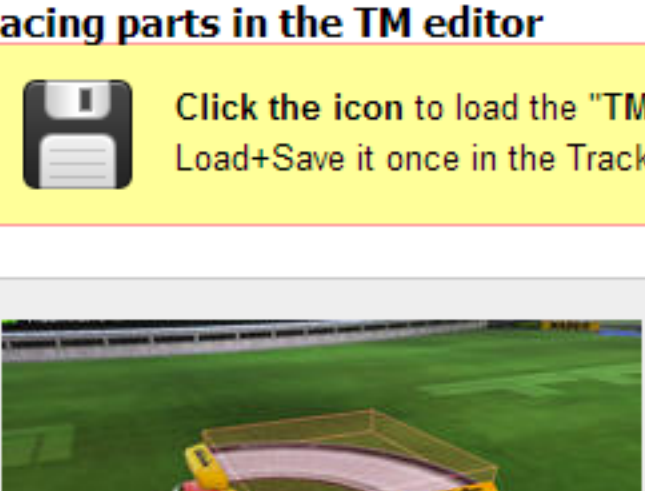
The french guys from [TM-Hexa forum](#) have provided us good tools, [\[ChallengeEdit \(Load\)\]](#) among them, which can decompress a track file, list all tiles, do the Hex editing all by itself and save the file again. A superb multi-functional tool do all necessary work for blockmixing.

ChallengeEdit can only edit tracks, which have been made with one of the local online profiles on your harddisk. So, before you can edit our example files, you must load them once in your track editor and save them again, so "you" become the author of the file.

Placing parts in the TM editor

 Click the icon to load the "**TMC-Blockmix-Example1**" and put it in your "Tracks/Challenges" folder.

Load+Save it once in the Trackmania editor, so it's useable for ChallengeEdit.

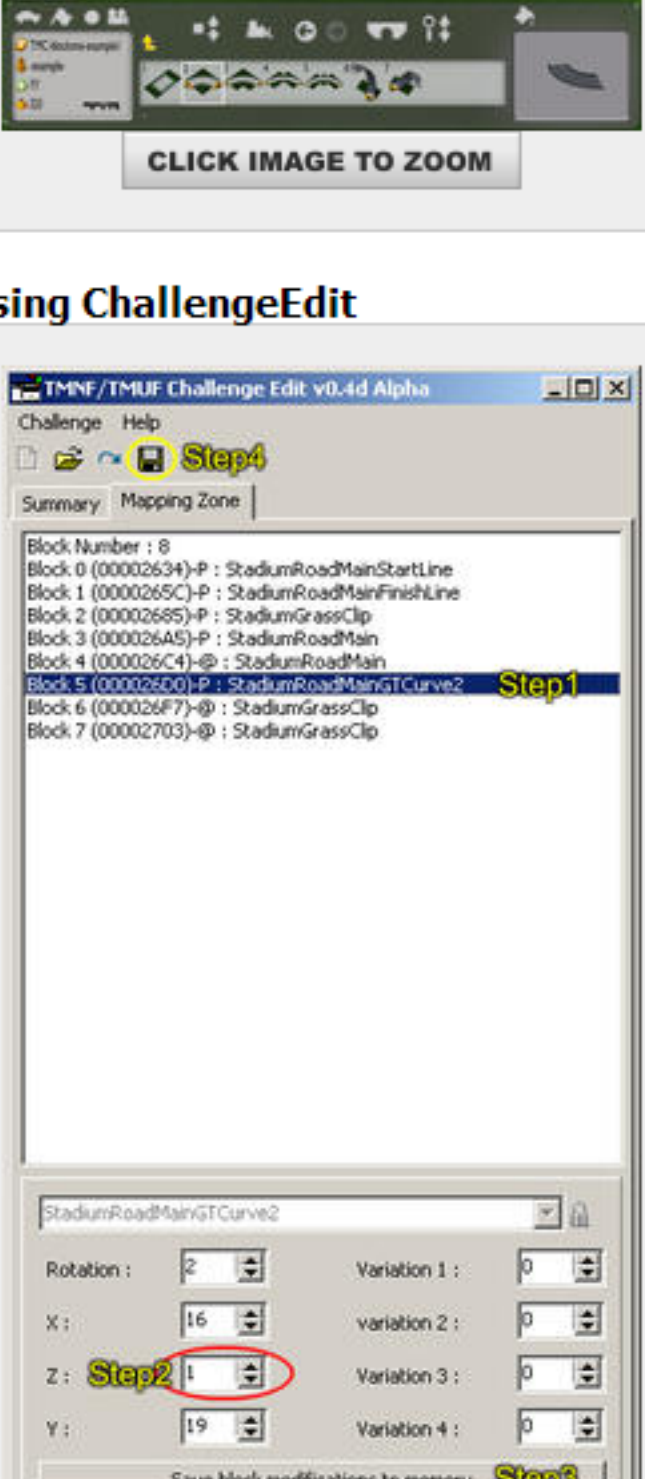


Open it in the TM editor. We want to create a real narrow curvy road.

Place the start ramp, some small edgy road and the finish line.

Now select the first available curve tile (2/1/2). To not interfere with the other road pieces or the finish top, we must place it 2 pieces above the ground (=at Y:3) Save the track !

Using ChallengeEdit



Start ChallengeEdit and Load the track.

Step1) "StadiumRoadMainGTCurve2" is the one we want to lower down to the ground, so select it with the mouse.

Step2) At the bottom you can now see all important values for the selected tile. Change the "Z" value from "3" to "1" (lowering it down by 2 cells), with the help of the arrows on the right (or with direct keyboard entry).

Note: ChallengeEdit uses XZY as coordinates (instead of XYZ) but that is not important...

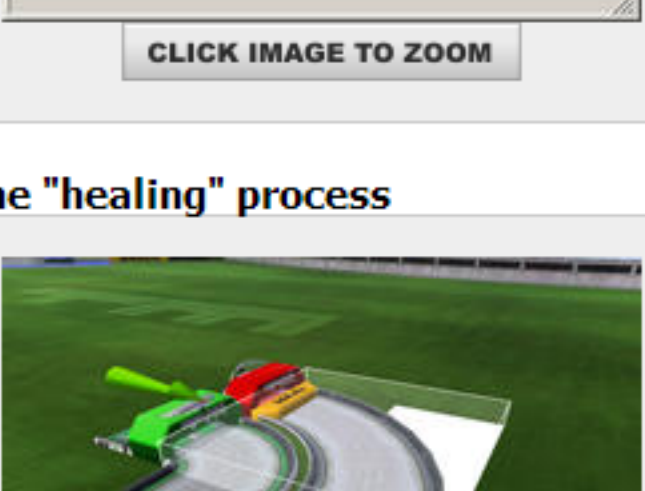
Look [HERE](#) for a detailed explanation why the values are exchanged)

Step3) Now press the "Save block modification..." button.

Step4) Press the "Save track" button.

The Hex editing is already done, go back to TM and load the track into the editor again.

The "healing" process



Back in the TM editor, we have a problem. We actually moved a "curve **ain** the air" down to the ground and now we have an air hole in the ground :-)

It's time to start a "**healing process**", using the TM editor to repair our broken mix.

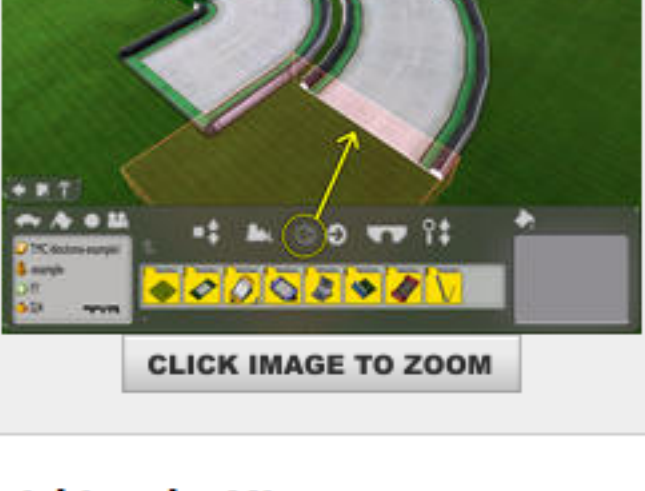
Click the "Erase" button and SELECT the curve piece.



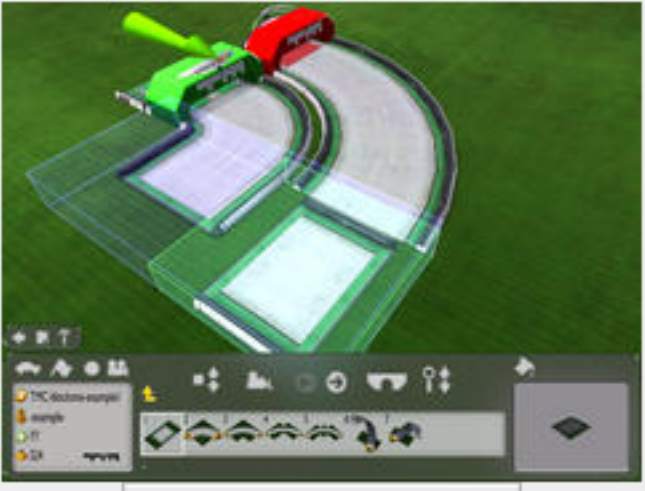
As soon as the curve piece is deleted, the floor is there again, but everything looks a bit scrambled...

Press the "**UNDO**" button and TM will try to re-install the curve piece, but with some intelligence, which means, that TM will now install the curve tile in a way, that it is properly connected to all tiles around it, also the floor...

Finishing the Mix

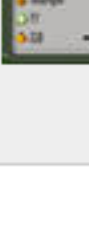


Now we can draw one more road piece (2/1/1) to connect the curve to the other road pieces on the left...



And now our narb curve construction is finished successfully !

This was a very simple example of "modern" blockmixing / hex-editing and basically, most mixing actions work like this...

 Click the icon to load the "**TMC-Blockmix-Example1-Mixed**" ... this is how the mixed track looks like, you just need to heal it, like described above.